

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

INTERNETOVÝ PORTÁL HUDEBNÍ SKUPINY

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

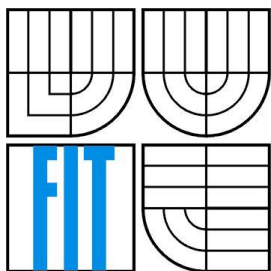
AUTOR PRÁCE  
AUTHOR

TOMÁŠ BOREK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## INTERNETOVÝ PORTÁL HUDEBNÍ SKUPINY

INTERNET PORTAL OF MUSICAL BAND

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

TOMÁŠ BOREK

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. JIŘÍ JAROŠ

BRNO 2008

## **Abstrakt**

Tématem této bakalářské práce je návrh a implementace Internetového portálu hudební skupiny v jazyce PHP, za využití databázového serveru MySQL a jeho nasazení do provozu. Internetový portál splňuje funkci prezentace hudební skupiny a členům hudební skupiny umožňuje dynamickou správu informací. Pro návštěvníky portálu je připravena možnost vkládání příspěvků do širokého spektra diskuzí, hlasování v anketách a možnost registrace do portálu.

## **Klíčová slova**

PHP, MySQL, WWW, hudební, skupina, Marshal

## **Abstract**

Main theme of this bachelor's thesis is to design and to implement Internet portal of musical band in PHP programming language using database server MySQL and to apply it of traffic. Internet portal realizes presentation duty and offers dynamic control of information for band members. There is provided an occasion for posting stories into wide spectrum of forums, voting in inquiries and registering into user's list in portal.

## **Keywords**

PHP, MySQL, WWW, musical, band, Marshal

## **Citace**

Borek Tomáš: Internetový portál hudební skupiny. Brno, 2008, bakalářská práce, FIT VUT v Brně.

# Internetový portál hudební skupiny

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jiřího Jaroše.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Tomáš Borek  
14. května 2008

## Poděkování

Touto cestou bych chtěl poděkovat vedoucímu práce Ing. Jiřímu Jarošovi za odborný dohled a vedení při vytváření této práce.

© Tomáš Borek, 2008

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah .....	1
Úvod .....	2
1 Historie internetu .....	3
2 Analýza .....	4
2.1 Specifikace .....	4
2.2 Programovací techniky .....	4
2.3 Programový model .....	5
2.4 Návrh databáze .....	7
3 Implementace .....	8
3.1 Engine .....	9
3.2 Uživatelé .....	12
3.2.1 Profil .....	12
3.2.2 Práva .....	12
3.2.3 Registrace .....	13
3.3 Kontexty .....	13
3.4 Sekce s jednou tabulkou .....	15
3.5 Sekce s více tabulkami .....	15
3.6 Textové sekce .....	16
3.7 Diskuze .....	17
4 Instalace portálu .....	18
4.1 Příprava .....	18
4.2 Instalace serveru Apache .....	19
4.3 Instalace interpretu PHP .....	19
4.4 Instalace databázového serveru MySQL .....	20
4.5 Nastavení a oživení webservru .....	20
4.6 Instalace internetového portálu .....	22
5 Závěr .....	24
Literatura .....	25
Seznam příloh .....	26

# Úvod

Předmětem této práce je vytvoření Internetového portálu hudební skupiny, přičemž základními stavebními kameny portálu jsou programovací techniky jazyka PHP a využití dotazovacího jazyka SQL. Internetový portál bude sloužit k několika účelům. Předně se jedná o webovou prezentaci hudební skupiny a tedy zveřejnění informací, tak, jak se v dnešní době očekává. Portál musí kromě veřejných informací zavádět vazbu zobrazovaných informací na uživatele, stejně jako umožnit rychlý a jednoduchý přístup k aktualizaci zobrazovaných informací.

Cílem práce je vytvoření prezentace s dynamickými údaji, které na jedné straně budou využívat jednoduchosti a střídmosti databázového serveru MySQL a na straně druhé bude postaven na robustním enginu implementovaném v jazyce PHP, který maximálně zapouzdří používání jazyka HTML a tím odizoluje vlastní programování portálu pouze na úroveň jazyka PHP.

Jedním ze stěžejních bodů je využití a spojení výhod jak modulárního tak objektového programování a také universálnosti jednotlivých částí portálu. Stejnou vahou je kladen důraz na spolehlivost výsledného produktu a na šíři jeho provozuschopnosti na různých verzích jazyka PHP a databázového serveru MySQL, nejen na nejnovějších.

Tento dokument se zabývá návrhem a vývojem portálu a řešením úskalí nebo omezení vzniklých během implementace. Dále pak popisuje postup zavedení portálu do provozu, včetně přípravy a vytvoření vhodného prostředí pro portál. V závěru poté shrnuje možnosti portálu, míru dodržení specifikace a možnosti budoucího rozšíření.

# 1 Historie internetu

Internet jako služba WWW [11], tedy World Wide Web byla poprvé spuštěna v roce 1990 jako předmět výzkumu. V té době byla tato služba založena na třech základních technologiích a to jazyku HTML, protokolu HTTP a URL. HTML je jazyk určený k zápisu dokumentů, což jsou základní stavební kameny WWW. V dnešní době se používá verze HTML 4.0 nebo modernější verze XHTML 1:1. Na to navazuje přenosový protokol HTTP, který je určen pro přenos výše zmíněných HTML dokumentů. Tento protokol ve své první verzi 0.9 představoval jen velmi malou množinu operací, ale v nynější verzi 1.0 a 1.1 se stává standardem, na jehož základě pracuje většina prohlížečů. Poslední technologií nezbytnou pro implementování služby WWW jsou URL což jsou ve své podstatě prvky jimiž lze jednoznačně adresovat umístění dokumentu v Internetu.

Velmi populární nadstavbou je skriptovací jazyk PHP [10]. První verzí PHP je malá sada skriptů napsaných v Perlu, které pro své soukromé účely sestavil Rasmus Lerdorf a nazval je Personal Home Page Tools. Další verzí, která vznikla na základě toho, že Rasmus zveřejnil zdrojové kódy uměla zpracovávat formuláře a byla to vlastně první verze podobná PHP tak jak je známe dnes.

V roce 1997 vzniká další verze PHP, tentokrát napsané již v jazyce C. Na této verzi spolupracuje řada nadšenců a nedlouho po té vychází PHP verze 3.0. Tuto verzi sestavili Andi Gutmans a Zeev Suraski, aby tak rozšířili zatím poněkud poddimenzovanou předchozí verzi.

Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. K poskytnutí pevné infrastruktury pro mnoho různých databází, protokolů a API koncovým uživatelům, přilákaly možnosti rozšíření PHP 3.0 také tucty vývojářů, kteří se připojili a vytvořili nové rozšiřující moduly. Toto byl nesporně klíč k obrovskému úspěchu PHP 3.0. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka. PHP 3.0 bylo oficiálně uvolněno v červnu 1998, poté co strávilo cca 9 měsíců ve veřejném testování.

V zimě 1998, krátce po oficiálním uvolnění PHP 3.0, začali Andi Gutmans a Zeev Suraski pracovat na přepsání jádra PHP. Cílem návrhu bylo zvýšit výkon pro složité aplikace a zlepšit modularitu kódové báze PHP. Nový engine, nazvaný Zend Engine úspěšně splnil cíle návrhu a byl uveden v polovině roku 1999. PHP 4.0, založené na tomto enginu a doplněné širokou škálou nových prvků, bylo oficiálně uvolněno v květnu 2000, necelé dva roky po svém předchůdci, PHP 3.0. K podstatně zvýšenému výkonu této verze přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

Dnes používají PHP (odhadem) stovky tisíc vývojářů a nainstalované PHP hlásí několik milionů serverů - tj. přes 20 % domén na Internetu. Vývojový tým PHP zahrnuje tucty vývojářů, stejně tak jako tucty dalších lidí, kteří pracují na projektech spojených s PHP, jako je PEAR a dokumentační projekt.

## 2 Analýza

V této kapitole je rozebráno jaké programovací techniky a programové vybavení bude použito při implementaci a co ovlivní jejich volbu. Dále je nastíněna problematika návrhu universálního enginu, který bude generovat a obsluhovat portál a bude specifický některými vlastnosti nebo použitím v případném jiném řešení internetové aplikace. Na závěr kapitoly je popsán návrh databázového modelu.

### 2.1 Specifikace

Internetový portál má přesně stanovené sekce, do kterých se zobrazované údaje budou řadit. Do hlavního menu spadají následující sekce.

Sekce Aktuality bude zobrazovat textové aktuality a automaticky vkládané aktuality. Sekce Akce bude zobrazovat seznam akcí podle roku pořádání akce nebo podle aktuálnosti akce. Sekce Foto bude obsahovat fotogalerie z akcí a sekce Diskuze umožní přístup k veřejné diskuzi. Sekce Playlist bude zobrazovat aktuální repertoár skupiny a umožní jeho zobrazení podle žánru jednotlivých písní. Sekce Ankety bude obsahovat seznam vložených anket. Sekce Členové zobrazí seznam fotografií členů skupiny. Kliknutím na fotografii se zobrazí profil zvoleného člena. V sekcích Historie a Kontakt se zobrazí historie skupiny potažmo kontaktní informace jako seznam textových bloků. Podobně se zobrazí seznam článků v sekci Články. V sekcích Download a Odkazy se zobrazí seznamy odkazů zařazených do kategorií. Poslední sekcí v hlavním menu je sekce Registrace, která pomocí formulářů umožní registraci uživatele, případně nabídne vygenerování zapomenutého hesla.

Ostatní sekce budou zařazeny v uživatelském menu a budou se zobrazovat pouze přihlášenému uživateli podle kategorie, do které spadá. Jedná o soukromou diskuzi a diskuzi pro členy skupiny, seznam uživatelů a informace o stavu portálu. K jednotlivým uživatelům se pak váží sekce s nastavením profilu a změny hesla. Pro členy skupiny bude k dispozici sekce, kde bude možné zadávat termíny např. pro zkoušky.

### 2.2 Programovací techniky

Jedním z cílů práce je možnost uvedení portálu do provozu na co možná nejširším spektru použitého programového vybavení. To se vztahuje i na výběr použitého interpretu PHP. Jako nejnižší použitelná verze se jeví verze 4.0. Od této verze je v PHP implementována důležitá změna co se týče knihovny pro práci s databázemi MySQL. Konkrétně je to možnost ukládání výsledků SQL dotazu do asociativního pole, kde je hodnota daného sloupce výsledku přístupná přes klíč, kterým je název sloupce. V předchozích verzích bylo umožněno získání výsledků tak, že klíčem bylo pořadí sloupce



specifikované v dotazu. Toto se jeví jako značně neefektivní v případě nutnosti provést změnu v dotazu, čímž se samozřejmě změní i pořadí a tedy klíče sloupců. Dalším způsobem, kterým bylo možné získat výsledek dotazu bylo získání pseudo-objektu, jehož proměnné nesly stejné názvy jako názvy sloupců v dotazu. Tato technika ovšem neumožní iteraci po sloupcích a tím výhodněji se jeví použití asociativního pole. V této chvíli je potřeba zvážit možnosti případného přechodu na jiný databázový server. V potaz lze vzít například databázový server FireBird, který je šířen jako freeware a je z 95% kompatibilní s databázovým severem InterBase. Lze tedy předpokládat jeho nasazení. Knihovní funkce pro InterBase nebo FireBird umožňují získávání výsledku dotazu do asociativního pole klíčovaného názvy sloupců od verze 4.3, tedy tato verze se uvažuje jako nejnižší použitelná.

Na druhou stranu je třeba rozebrat provoz portálu na novějších verzích PHP, konkrétně na aktuální verzi 5. Narozdíl od verze 4, vyšší verze disponují plnou podporou objektového programování. Doplnují chybějící možnost použití destrukturu objektu a umožňují použití modifikátorů viditelnosti metod a vlastností objektu. Ovšem z pohledu implementace portálu nevyvstává potřeba použití těchto možností.

Další důležitou změnou ve vyšších verzích PHP je změna syntaxe jazyka. Od verze 6 nelze předávat parametr funkce odkazem zvenčí funkce, ale předání parametru odkazem lze provést pouze definicí v hlavičce funkce.

Při dodržení výše uvedených pravidel lze předpokládat funkčnost portálu na PHP od verze 4.3 až po verzi 5 a vyšší.

Narozdíl od PHP jsou změny mezi verzemi MySQL spíše aditivního charakteru. Vzhledem k tomu, že se nepředpokládá využití vyšších programovacích technik v SQL jako je použití transakcí nebo uložených procedur, je jako referenční verze MySQL zvolena stabilní verze 4.1 a předpokládá se provoz portálu i na novějších verzích MySQL. Vzniká zde ovšem problém s případným přechodem na jiný databázový server, např. FireBird. Mezi těmito servery dochází k odlišnostem v syntaxi jazyka. To je ovšem neměnný fakt, tedy volba verze zůstává stejná.

Jako poslední zbývá zvážení použití dalších programovacích jazyků nebo prvků jako je JavaScript, Java aplety nebo Flash animace, které by jistě zvýšili komfort používání portálu či jeho možnosti. Proti těmto prvkům ovšem mluví fakt, že uživatel musí doinstalovat do svého prohlížeče příslušnou podporu. Nelze přehlížet také některá bezpečnostní opatření, jakými je např. zákaz provádění JavaScriptu. Z těchto důvodů nebude žádný z těchto prvků použit.

## 2.3 Programový model

Základním kamenem programového modelu je kombinace modulárního a pseudo-objektového programování. V implementaci by mělo být využito objektového programování v podobě, v jaké je poskytuje PHP verze 4.3. Případné chybějící a nenahraditelné možnosti objektového programování

nahradí poněkud „vyjíměčné“ jazykové konstrukty PHP jako jsou volání funkcí a metod pouze jejich textovým názvem (ne tedy volání standardním voláním funkce či metody).

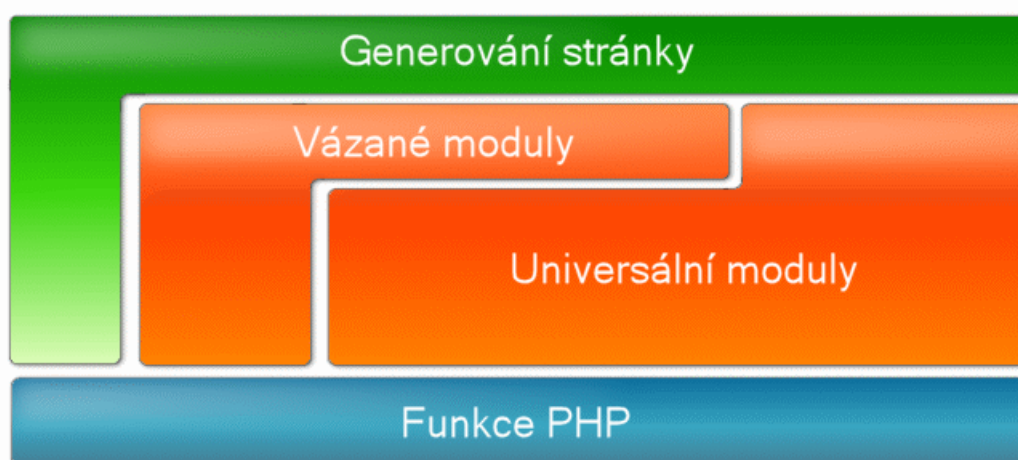
Na druhé straně modulární programování umožní členění částí kódu do logických bloků. S tímto se pojí návrh enginu, který umožní v reálném čase jednotlivé moduly připojovat, vytvářet instance tříd definovaných uvnitř modulů anebo tyto moduly odebírat a tedy uvolňovat zdroje modulem využívané. Návrh této modulární struktury umožní implementovat jednotlivé části kódu podle předem stanovené šablony a tím se stane používání modulů v dostatečné míře universální. Z toho vyplývá, že v případě potřeby přidání nových modulů, které nejsou zatím v této chvíli známe, postačí dodržet předepsanou šablonu a nové moduly budou funkční.

Důležitou funkcí enginu je na prvním místě oddělení psaní klasického HTML kódu od prokládání HTML kódu voláním částí PHP skriptů. Každý skript, jehož výsledkem bude výstup do HTML dokumentu, bude používat pouze funkce, které toto zprostředkují.

Od toho se odvíjí i logické rozdělení funkcí do modulů. Základním modulem bude seznam funkcí zapouzdřující výstup HTML tagů a pomocných funkcí. Další modul bude zpracovávat rozsáhlejší konstrukce HTML jako jsou tabulky nebo formuláře. V případě formulářů se očekává rozšířená funkčnost a to na zpracování údajů obsažených ve formuláři. Do oddělených modulů se zařadí funkce nebo třídy zapouzdřující práci s databází, generování šablony HTML dokumentu a práce s používanými formáty jako je formát „ini“ souboru.

Výše zmíněné moduly tvoří universální jádro enginu, které bude použitelné nezávisle na kategorii či zařazení portálu nebo webové prezentace. Doplněním onoho jádra enginu budou takzvané vázané moduly a to ve smyslu toho, že pro svoji činnost již vyžadují základní moduly nebo jiné zdroje. Tyto moduly jsou vázané na konkrétní implementaci portálu a zapouzdřují práci s uživateli, logické členění portálu a funkce specifické právě pro tento portál.

Na obrázku 2.1 je nastíněno jakým způsobem bude tedy probíhat vlastní generování stránky.



Obrázek 2.1: Generování stránky.

Posledním důležitým aspektem programového modelu je maximální možné oddělení grafického návrhu portálu od programového vybavení. Tohoto lze dosáhnout umístěním skupin HTML tagů specifikujících vzhled mimo programové řešení. V kombinaci s použitím definice vzhledu v kaskádovými styly bude splnění tohoto bodu dostačující.

## 2.4 Návrh databáze

V kapitole Programovací techniky bylo uvedeno jaká verze databázového serveru MySQL byla zvolena a jaká to přináší případná omezení. Z toho vyplývá, že návrh databáze bude založen na co nejjednodušším modelu. To podporuje i povaha dat, která budou v databázi udržována. Z části se bude jednat o jednoduché tabulky, které nejsou s dalšími nijak provázány. Obsah těchto tabulek bude použit jako lineární seznam. Rozdělování nebo kategorizace je plně proveditelná pomocí SQL dotazů. Tyto tabulky pokryjí údaje sekcí Aktuality, Akce, Playlist.

Další skupina tabulek je velmi podobná předchozí, s tím rozdílem, že je nutné označit jednotlivé záznamy podle podkategorie do které spadají. Na základě tohoto příznaku lze potom opět rozdělování dat provést za pomoci SQL dotazů. Do této skupiny spadají tabulky sekcí Odkazy, Kontakt a Download.

Sekce Členové, Historie a Články mají společnou vlastnost a to, že zobrazují rozsáhlá textová data. V návrhu pro ně tedy postačí jediná tabulka, která podobně jako v předchozím případě bude obsahovat identifikaci, které sekci příslušný záznam náleží.

Tabulky s vazbou 1:N pokryjí potřeby sekce Foto, kde se ke každé fotogalerii váže sada fotografií, a také poslouží v sekci Ankety, kde se ke každé anketě váže sada možností hlasování a v neposlední řadě také .sekcí Termíny, kde se k termínu váže vyjádření uživatele.

Speciálním případem je pak návrh tabulky pro diskuzi. V portálu se vyskytuje hned několik druhů diskuzí a to samostatné diskuze nebo diskuze vztahující se buď k záznamu jiné tabulky nebo k obecně nějaké sekci. Tyto diskuze musí umožnit zadávání odpovědí na příspěvky, s výjimkou diskuze soukromé, u níž je tato vlastnost bezpředmětná. Je třeba navrhnout jedinou tabulku, která bude mít výše zmíněné vlastnosti a každý záznam bude snadno přiřaditelný k typu diskuze.

Poslední tabulkou, která se váže k sekci je tabulka uživatelů. Tabulka eviduje veškeré údaje o uživateli, kategorizuje uživatele do role, udržuje informace o jeho aktivitě. Předpokládá se, že na tuto tabulky se mohou odkazovat jiné tabulky, předně na základě potřeby jednoznačně identifikovat uživatele.

V pozadí zůstává tabulka pro počítadlo přístupů. Na jednu stranu lze počítadlo navrhnout daleko jednodušeji, ale na stranu druhou, evidování každého přístupu a informací o přístupu umožní výpočet statistik a také např. rozpoznání přístupů způsobených pouze přechodem na jinou stránku portálu.

### 3 Implementace

Vzhledem k tomu, že konkrétní implementace je velmi závislá na přesných požadavcích ovládání portálu, je v této kapitole rozebíráno i ovládání, které svým charakterem navozuje způsob implementace nebo naopak.

Na obrázku 3.1 je základní rozvržení stránek. V záhlaví je umístěno logo hudební skupiny, pod ním pak hlavní menu portálu a dále hlavní zobrazovací prostor rozdělený na dvě části. Vlevo se zobrazuje obsah zvolené sekce a pravou část vyplňuje informační sloupek.

Informační sloupek využívá informací z některých sekcí a zobrazuje zpravidla nejaktuálnější informaci z dané sekce. Nejvýše je zobrazena akce, která se koná nejbližší aktuálnímu datu a to samozřejmě datu směřujícímu do budoucna. Pod aktuální akcí se nachází kalendář akcí. Tučně je zvýrazněno aktuální datum a podbarvením jsou zvýrazněny plánované akce. Kliknutím na barevně zvýrazněné datum se pod kalendářem zobrazí informace o akci na dané datum plánované.

Dále následuje nejnovější zveřejněná anketa, poté přihlašovací formulář a seznam sponzorů. Jako poslední uzavírá informační sloupec statistika přístupů. Jednotlivé části informačního sloupku budou vysvětleny vždy u sekce, ke které náleží.

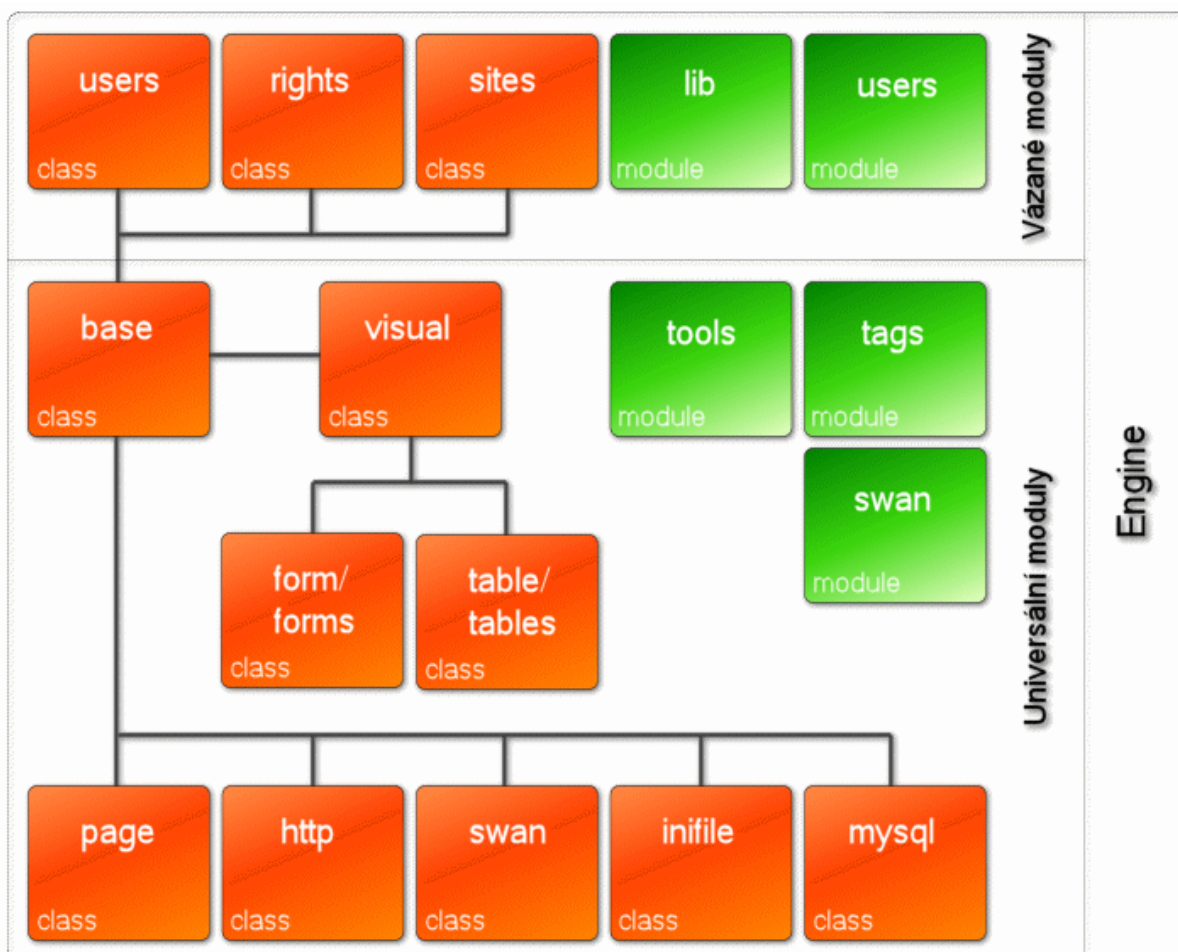


Obrázek 3.1: Základní rozvržení stránky.

## 3.1 Engine

Engine, tedy výkonné jádro, je systém modulů, které jsou řízeny z jednoho místa. Základním modulem je modul Swan. Jedná se o skript, který je nutné jako jediný vložit, aby byl engine dostupný. Vložением tohoto modulu se provede inicializace řídicí třídy SwanClass, pomocí které lze libovolně načítat a uvolňovat další moduly. Jak již bylo řečeno v kapitole Analýza, moduly lze rozdělit na universální a vázané. Universální moduly nepotřebují žádné další prostředky ke své funkci, kdežto vázané jsou založené buď na universálních modulech nebo potřebují jiné prostředky, které jsou zpravidla k dispozici až u konkrétní implementace. V případě tohoto portálu vázané moduly pokrývají funkci správy uživatelů, jejich práv, správy logiky generování stránek a také zapouzdřují sadu funkcí pro portál.

Na obrázku 3.2 je znázorněna logika spojení modulů. Jedná se o moduly, které implementují třídu nebo zapouzdřují sadu funkcí (na obrázku jsou to bloky *class* a *module*).



Obrázek 3.2: Blokové schéma enginu.

Po vložení základního modulu Swan do skriptu se jeho provedením vytváří objekt Swan, který má metody *LoadModule* a *LoadExternalModule*. Tyto dvě metody umožní připojení dalšího

universálního resp. vázaného modulu. Funkce také implementují volání inicializačního bloku modulu, ve kterém se zpravidla vytváří instance třídy obsažené v modulu, která se v té chvíli stává dostupnou pro zbytek skriptu. Volání této inicializační rutiny se provádí za použití PHP funkce *call\_user\_func*, jejímž parametrem je název volané funkce v podobě řetězce. Tento princip dává možnost sestavovat názvy volaných funkcí až za běhu a toho je právě využito. Název inicializační rutiny je sestaven podle vzoru *\_moduleInit\_*, kde se vloží název modulu na místo řetězce „module“. Podobným způsobem lze za pomoci metody *UnloadModule* a funkce *\_moduleDone\_* uvolnit alokované prostředky modulu.

Jak již bylo zmíněno, většina modulů implementuje právě nějakou třídu. Základní třídou je třída *BaseClass*, která implementuje chybějící objektovost PHP verze 4. Tuto potřebu uspokojí implementace pseudo-destruktoru, protože destruktor jako takový není v PHP 4 podporován. Celkově jde o dvě metody třídy *BaseClass* a to metoda *Destroy* a *Release*. Metoda *Destroy* provede uvolnění všech interních členů a nastaví instanci třídy do stavu, kdy čeká na uvolnění funkcí *unset* nebo na dokončení skriptu, kdy je instance uvolněna interpretem. Je důležité v každé zděděné třídě tuto metodu správně implementovat, tedy doplnit kód, který uvolní členy zděděné třídy a poté zavolat metodu *Destroy* rodičovské třídy. Pro uvolnění třídy z paměti se volá ovšem metoda *Release*, která před uvolněním ověří, zda již metoda *Destroy* nebyla volána.

Další třídou, která sama o sobě slouží jako šablona ostatním třídám, je třída *VisualClass*. Jak je patrné z názvu, jedná se o třídu, jejíž zdědění se předpokládá v případech, kdy třída implementuje HTML entity. Základní metodou této třídy je metoda *Tag* která standardizuje tisk HTML tagů do dokumentu. Její význam spočívá v přehlednosti výsledně vygenerovaného HTML kódu. Ke svázání HTML entit a kaskádových stylů slouží metoda *SetDefaultCssClasses*, která je přetížena v každé zděděné třídě a tím definuje výchozí styl, který bude při generování použit.

Všechny třídy v modulech lze nyní rozdělit na visuální a nevisuální. Do nevisuálních se řadí třída *PageClass*, která implementuje generování hlaviček HTML dokument, třída *HttpClass*, která zapouzdřuje práci se superglobálními poli GET, POST a SESSION, dále pak třída *IniFile*, která implementuje práci s řetězcí nebo soubory ve formátu „ini“ souboru a nakonec třída *MySQLClass*, která zapouzdřuje funkce *mysql\_\** pro práci s databázovým serverem MySQL.

Do seznamu visuálních tříd potom spadá dvojice tříd *TableClass* a *TablesClass*, které dohromady vytváří manažer pro generování tabulek a dvojice tříd *FormClass* a *FormsClass*, které dohromady vytváří manažer pro generování formulářů. Oproti ostatním třídám, které mají svůj význam definovaný už názvem, se u třídy *FormClass* skrývá ještě další funkcionality než jen vygenerování kódu. Tato třída implementuje generování polí formulářů podle jejich obsahu (což může být text, číslo, datum, čas, seznam a jiné) a kontrolu správnosti těchto údajů. Třída poskytuje také tři pseudo události, které nastávají v životním cyklu formuláře. První událost *OnLoad* je vyvolána před vygenerováním formuláře a umožní tak přednaplnit hodnoty polí. Druhá událost *OnValidate* je vyvolána před zpracováním údajů ve formuláři a slouží k druhotné kontrole údajů a třetí událost

*OnSubmit*, která je vyvolána po úspěšných kontrolách údajů po odeslání formuláře některým z tlačítek typu submit.

Použití enginu ilustruje obrázek 3.3. Tento kód je obsažen v indexovém skriptu portálu odkud probíhá řízení generování všech stránek. Kód je rozdělen do očíslovaných bloků jejichž význam je následující. Blok č.1 provádí inicializaci enginu a načtení všech universálních a vázaných modulů, které jsou potřeba pro správnou funkci portálu. Blok č.2 demonstruje použití vázaného modulu Sites, který generuje hlavní i uživatelské menu portálu a obsluhuje přepínání stránek. Blok č.3 konfiguruje a navazuje připojení k MySQL serveru. Blok č.4 nastaví hlavičky HTML dokumentu, blok č.5 pak volá skript, který generuje obsah příslušné stránky a poslední blok č.6 provede úklid paměti.

```
require_once 'swan/swan.php';

$Swan->LoadModule('http');
$Swan->LoadModule('page');
$Swan->LoadModule('tags');
$Swan->LoadModule('tools');
$Swan->LoadModule('mysql');
$Swan->LoadModule('forms');
$Swan->LoadModule('tables');
$Swan->LoadModule('inifile');
$Swan->LoadExternalModule('sys', 'lib');
$Swan->LoadExternalModule('sys', 'users');
$Swan->LoadExternalModule('sys', 'user.class');
$Swan->LoadExternalModule('sys', 'sites.class');
$Swan->LoadExternalModule('sys', 'rights.class');

$http->OpenSession();

$Sites->RegisterMenuSite("aktualne", "aktuálně", 87, 30);
$Sites->RegisterMenuSite("akce", "akce", 53, 30);
$Sites->RegisterMenuSite("foto", "foto", 43, 30);
$Sites->RegisterMenuSite("playlist", "playlist", 69, 30);
$Sites->RegisterMenuSite("diskuze", "diskuze", 73, 30);
$Sites->RegisterMenuSite("ankety", "ankety", 69, 30);
$Sites->RegisterMenuSite("clenove", "členové", 79, 30);
$Sites->RegisterMenuSite("historie", "historie", 70, 30);
$Sites->RegisterMenuSite("download", "download", 96, 30);
$Sites->RegisterMenuSite("clanky", "články", 67, 30);
$Sites->RegisterMenuSite("registrace", "registrace", 94, 30);
$Sites->RegisterMenuSite("kontakt", "kontakt", 76, 30);
$Sites->RegisterMenuSite("odkazy", "odkazy", 71, 30);
$Sites->RegisterUserSite("web", "web", ADMIN);
$Sites->RegisterUserSite("uzivatele", "uživatelé", MEMBER);
$Sites->RegisterUserSite("termíny", "termíny", MEMBER);
$Sites->RegisterUserSite("clenzpravy", "zprávy kapely", MEMBER);
$Sites->RegisterUserSite("soukzpravy", "soukromé zprávy", USER);
$Sites->RegisterUserSite("profil", "profil", USER);
$Sites->RegisterUserSite("heslo", "heslo", USER);
$Sites->RegisterBlindSite("prohlizec", "prohlížení fotek");

$MySQL->SetConf(LOCALHOST, 'localhost', '', '', 'marshal');
$MySQL->SetConf(SERVER, 'mysql', 'uzivatel', 'heslo', 'databaze');
$MySQL->Connect();

$page->Set('title', '.: marshal .: $Sites->Sites[$Sites->SiteId]['title']. ' .:');
$page->Set('author', 'Tomáš Borek, xborek03, FIT VUT 2008');
$page->Set('keywords', 'marshal, kapela, zábavová');
$page->AddCascade("css/$UseCSS.css");

$script = 'sites/'. $Sites->SiteId. '.php';
include $script;

$http->CloseSession();
$Swan->FreeModules();
$Swan->Destroy();
```

Obrázek 3.3: Použití enginu ve skriptech.

## 3.2 Uživatelé

K úplnému vysvětlení funkce portálu je nejdříve nutné vysvětlit princip uživatelského přístupu a kontextů. Začneme tedy profilem uživatele.

### 3.2.1 Profil

O každém uživateli jsou evidovány údaje informativního charakteru a to jméno, příjmení, adresa webových stránek a ICQ číslo. O zveřejnění těchto údajů rozhoduje sám uživatel tím, že je vyplní nebo nevyplní. Tyto údaje jsou tedy nepovinné.

Dále jsou evidovány povinné údaje a to přezdívka, heslo, e-mail a statistické údaje: datum a čas registrace, datum a čas posledního přihlášení, počet přihlášení, IP adresa stroje, ze kterého se uživatel přihlásil a zda je uživatel právě přihlášen.

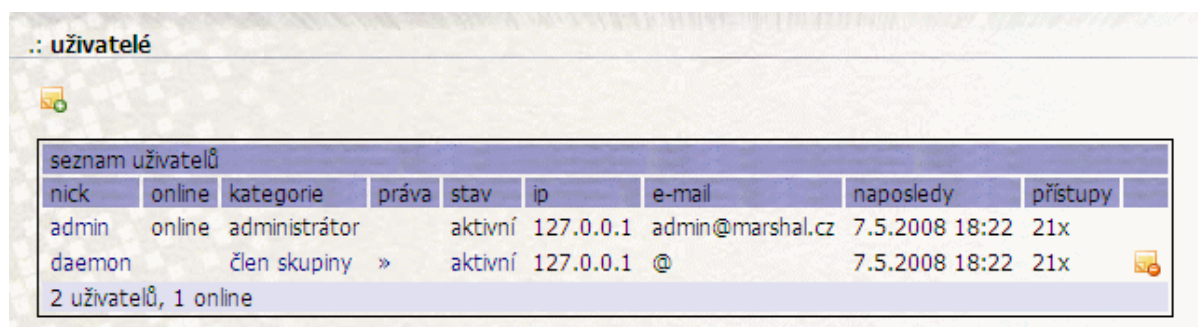
### 3.2.2 Práva

Posledními evidovanými údaji jsou kategorie uživatele, stav a práva uživatele. Kategorie rozděluje uživatele na tři skupiny, běžný uživatel, člen skupiny a administrátor. Běžný uživatel je vytvořen automaticky zaregistrováním nebo v může být v krajním případě vytvořen administrátorem.

Běžný uživatel má výhodu oproti ostatním návštěvníkům, že je jeho přezdívka chráněna před zneužitím a má možnost zobrazovat svoje informativní údaje. Dále pak je mu přístupná soukromá diskuze, ve které může skrytě komunikovat s ostatními uživateli.

Kategorie Člen skupiny je určena, jak již vyplývá z názvu, pro členy skupiny. U této kategorie jako jediné jsou využita práva uživatele. Pro každou sekci je vyhrazeno právo administrace sekce. Toto právo přiděluje administrátor.

Poslední kategorie, tedy administrátor poskytuje uživateli veškerá práva na administraci, včetně správy uživatelů. Na obrázku 3.4 je vyobrazeno rozhraní pro správu uživatelů, které je k dispozici administrátorům.



nick	online	kategorie	práva	stav	ip	e-mail	naposledy	přístupy
admin	online	administrátor		aktivní	127.0.0.1	admin@marshal.cz	7.5.2008 18:22	21x
daemon		člen skupiny	>	aktivní	127.0.0.1	@	7.5.2008 18:22	21x

2 uživatelů, 1 online

Obrázek 3.4: Rozhraní pro právu uživatelů.



Zbývá osvětlit údaj evidovaný jako stav. Tento údaj nabývá čtyřech hodnot. Stav „registrace“ je přidělen uživateli, který se zaregistroval v sekci Registrace a plní účel odlišení od ostatních uživatelů. Tento stav se změní na další možný stav, stav „aktivní“. To je normální stav, kdy uživatel může provádět operace, které přísluší jeho kategorii. Další stav se nazývá „zakázaný“. Je to stav, kdy se uživatel nemůže přihlásit, přičemž při pokusu o přihlášení je informován o tom, že zřejmě porušil některé z pravidel stanovených administrátorem a u něj může žádat o vysvětlení či převedení zpět do aktivního stavu. Poslední stav nazvaný „vyřazený“ je stav kdy je z nějakého důvodu nutné zakázat uživateli přístup a to tak, jakoby nebyl vůbec registrován. Vhodným příkladem pro použití toho stavu, je zablokování vybraných přezdivek nebo e-mailů.

Princip kategorií je založen na jednom či více administrátorech, kteří mají zodpovědnost za chod portálu. Ti určují pravidla, podle kterých používají uvedené možnosti změny kategorií nebo stavu. Změny kategorií i stavu se provádí kliknutím na danou kategorii nebo stav uživatele. Hodnota po každém kliknutí posune na další v řadě.

### 3.2.3 Registrace

Pro úplnost práce s uživateli je na místě osvětlit registraci uživatelů. Ta se provádí v sekci Registrace. Při registraci je nutné zadat do formuláře zvolenou přezdívku a platnou e-mailovou adresu. Registraci lze provést pokud oba údaje jsou v databázi pouze jednou. Po vyplnění těchto údajů se zašle na zadanou e-mailovou adresu e-mail s informací o registraci, v němž je uvedeno náhodně vygenerované heslo.

Heslo si uživatel po prvním přihlášení může změnit. Heslo je ukládáno do databáze v šifrované podobě, aby se předešlo zneužití. Po úspěšném odeslání e-mailu se uloží údaje do databáze a stav uživatele se nastaví na „registrovaný“. Pokud se v tomto stavu uživatel nachází déle jak 7 dní, je automaticky z databáze vyřazen. V opačném případě se po prvním přihlášení automaticky převede stav na „aktivní“. Prvním přihlášením se také ověří pravost e-mailové adresy uvedené při registraci. Na tuto adresu bude zasláno nově vygenerované heslo v případě, že stávající heslo uživatel zapomene a v sekci Registrace o něj požádá.

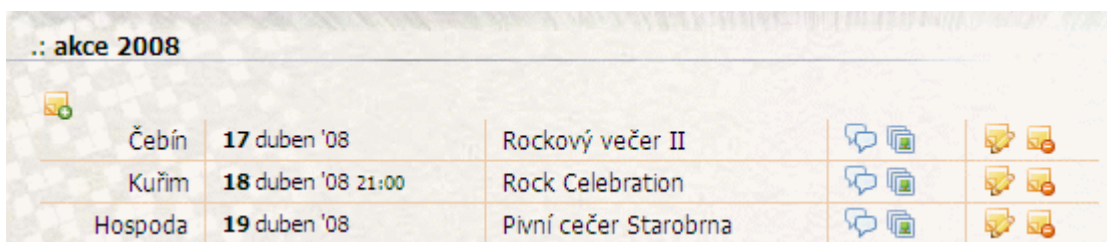
## 3.3 Kontexty

Portál je navržen tak, aby bylo administrační rozhraní implementováno přímo v prezentaci a nebylo tedy nutné používat druhotnou aplikaci k administraci stránek. To vede k několika stavům, ve kterých se portál může nacházet. Stav kdy není přihlášen žádný uživatel — veřejný kontext, stav kdy uživatel přihlášen je — uživatelský kontext a speciální stav, kdy je přihlášeným uživatelem přímo administrátor nebo člen skupiny — administrační kontext.

Pokud se portál nachází ve veřejném kontextu, je zobrazen pouze jako webová prezentace. Přihlášením uživatele se portál dostává do uživatelského nebo administračního kontextu podle

kategorie a práv, které danému uživateli náleží. V obou těchto případech se na první místo do pravého informačního sloupce přidá uživatelské menu. Přihlášený uživatel může přes toto menu editovat svůj profil, změnit svoje heslo a přidávat příspěvky do soukromé diskuze. Pokud přihlášený uživatel spadá do kategorie „člen skupiny“, zobrazí se v menu navíc položky, které umožní přidávat příspěvky do diskuze určené výhradně členům skupiny, registrovat termíny a zobrazit seznam uživatelů. Pokud spadá přihlášený uživatel do kategorie „administrátor“, umožňuje mu seznam uživatelů zároveň manipulaci s uživateli, tak jak bylo uvedeno v kapitole Práva. Administrátor navíc vidí položku menu „web“, která zobrazí údaje o stroji na němž portál běží, údaje o stavu enginu a na závěr tak funkci pro nastavení data poslední aktualizace portálu.

Z výše uvedeného vyplývá, že rozdíl mezi uživatelským a administračním kontextem nelze vyvodit pouze z kategorie uživatele, ale je nutné vzít v potaz také práva uživatele. Pak se tedy portál nachází v administračním kontextu v případě, že je přihlášen administrátor nebo člen kapely, který má pro danou sekci nastaveno právo k administraci.



Obrázek 3.5: Ikony pro základní administrační operace — přidávání, úprava a odebírání záznamu.

Význam administračního kontextu je pak přímo spojen s integrací administračního rozhraní přímo do portálu. Vlastní administraci se rozumí základní operace: přidávání, úprava a odebírání záznamů v příslušné tabulce nebo operace vynucené povahou údajů, např. skrývání nebo obnova příspěvků v diskuzi. Tyto operace jsou u každého záznamu příslušné tabulky reprezentovány ikonou, tak jak je patrné na obrázku 3.5 a 3.6.



Obrázek 3.6: Ikony pro operace skrytí a obnovení příspěvku.

## 3.4 Sekce s jednou tabulkou

Údaje v sekcích jsou z velké části tabulárního charakteru, nejsou nijak provázány a tudíž jsou také uloženy pro každou takovou sekci v jediné tabulce. Operace definované nad takovou tabulkou jsou přidávání, úprava a odebrání záznamu. Řazení záznamů je nastaveno intuitivně nad jedním nebo více sloupci podle povahy údajů. Například v sekci Akce se řazení provádí podle data, naproti tomu v sekci Playlist podle autora a názvu písně. Řazení a další kategorizace údajů se provádí na úrovni SQL dotazu. Další kategorizací se rozumí rozdělení seznamu akcí podle roku kdy se akce konala nebo ve vztahu k aktuálnímu datu, tedy akce uskutečněné a akce plánované.

Editace údajů se provádí pomocí formulářů a jejich napojení na tabulky tak, jak to umožňuje engine. Sekce s jednou tabulkou jsou sekce Aktuálně, Akce, Playlist, Download, Kontakt a Odkazy. Kompletní popis tabulek je uveden v příloze 2.

## 3.5 Sekce s více tabulkami

Velmi podobným způsobem jako v předchozím případě jsou uložena data pro sekce Foto, Ankety a Termíny s tím rozdílem, že výsledná zobrazená data se získávají spojením dvou tabulek, z nichž je vždy jedna nadřazena druhé, jde tedy o vazbu 1:N.

V případě sekce Foto je nadřazenou tabulkou tabulka galerií a podřazenou je tabulka fotografií. Plnění dat do sekce Foto probíhá poněkud odlišně než je tomu u ostatních tabulek. Tabulky totiž kopírují adresářovou strukturu galerií, což přináší výhodu při kopírování fotografií na server. Není nutné za pomoci formulářů nahrávat na server fotografie po jedné, ale pomocí služby *ftp* se vytvoří adresář galerie a fotografie se tam tak nahrají najednou. Při zakládání galerie v portálu se vždy galerie sváže s adresářem a jednotlivé fotografie jsou pak zpracovány automaticky. Jednorázové zpracování všech fotografií zahrnuje jejich umístění do galerie, kontrolu duplikátů a vytvoření náhledu.

V případě sekce Ankety je nadřazenou tabulkou tabulka anket k níž se váže tabulka možností. Administrační rozhraní umožňuje standardní operace přidání, úpravy a odstranění jak na úrovni ankety, tak na úrovni možnosti ankety.

Obdobně pracuje i sekce Termíny, která navíc přebírá údaje z tabulky uživatelů, konkrétně uživatele spadající do kategorie Člen kapely. Seznam členů kapely je nabídnut při vytváření nového termínu a slouží k výběru členů, kterých se termín týká. Vybrané členy nelze posléze měnit. Každý člen se ke zvolenému termínu potom vyjádří výběrem možnosti Ano nebo Ne.

## 3.6 Textové sekce

Mezi textové sekce se řadí sekce Členové, Historie a Články z toho důvodů, že obsah těchto sekcí je tvořen textovými bloky. Všechny tyto bloky jsou uchovávány v jedné tabulce a pomocí příznaku jsou přiřazeny k sekcím. Bloky jsou zobrazovány jako HTML kód, lze tedy využít HTML značky.

Speciálním případem je blok pro sekci Členové, který je pro sekci jediný a je zobrazován s jiným formátováním. Předpokladem je, že v této sekci se budou údaje aktualizovat minimálně a je tedy možné použít zvláštního formátu údajů. Základem formátu je formát jednoduchého „ini“ souboru. Hlavní sekce tohoto „ini“ souboru, nebo přesněji „ini“ bloku je sekce *členové*. Klíče v těchto sekcích jsou stejné a jejich hodnoty vyjmenovávají přezdívky jednotlivých členů hudební skupiny.

```
[členové]
člen = Tom
člen = Věnec
člen = Krab
člen = Gabo
člen = Cleposh
```

Při zobrazování se předpokládá existence sekce, která bude nazvána stejně jako přezdívka. V této sekci se nastaví informace o členu skupiny a to způsobem kdy se zobrazí klíč i hodnota ve formě tabulky. Lze si vynutit i formátování hodnot sekce a to tak, že se za klíč sekce přidá znak „|” a za něj některý z vybraných typů, tak jak to demonstruje následující výpis:

```
[Věnec]
ikona|ikona = icon_jura.gif
jméno = Jiří Jaroš
narozen|datum = 22.7.1980
obrázky|obrázky = foto_jura.jpg, foto_jura_2.jpg, foto_jura_3.jpg,
foto_jura_4.jpg
icq|icq =
email|email =
bydliště = neznámo
nástroj|text = Věnec_nástroj
```

Lze zvolit z vybraných typů *datum*, který hodnotu klíče naformátuje jako datum, *icq* který přidá ikonu stavu ICQ klienta a hodnotu klíče považuje za ICQ číslo, dále *email*, což provede vytvoření odkazu s direktivou „mailto:“. Dále je k dispozici formát *obrázky* který očekává jako hodnotu seznam názvů souborů s fotografiemi oddělených čárkou. Tyto fotografie se zobrazí v profilu člena skupiny. Obdobně pracuje formát *ikona* s tím rozdílem, že se fotografie zobrazí v seznamu členů. Jako výchozí adresář pro obrázky je nastaven adresář *members* v kořenovém adresáři portálu. Poslední formát *text* očekává jako hodnotu název další sekce, která se ovšem nezpracovává jako dvojice klíč-hodnota, ale jako prostý text, např. takto:

```
[Věnec_nástroj]
basa: Ibanez <b>K5</b>
aparát: Hartke HA3500
struny: EBS Titanium
```

Výsledek formátování je vyobrazen na obrázku 3.7.



Věvec	
jméno	Jiří Jaroš
narozen	22 červenec '80
obrázky	22 červenec '80
icq	
email	<a href="mailto:jirka@kapela-marshal.cz">jirka@kapela-marshal.cz</a>
bydlište	neznámo
nástroj	basa: Ibanez <b>K5</b> aparát: Hartke HA3500 struny: EBS Titanium

Obrázek 3.7: Formátování v sekci Členové.

## 3.7 Diskuze

Diskuze je z pohledu implementace nejzajímavější. Portál totiž obsahuje hlavní diskuzi, diskuzi pro členy kapely, soukromou diskuzi, diskuzi k playlistu a neomezený počet diskuzí ke galeriím. To vedle k vytvoření jediné tabulky, která universálně pojme všechny tyto diskuze a bude dále rozšiřitelná.

Tabulka tedy obsahuje základní sloupce jako čas vložení příspěvku, obsah příspěvku, vazbu na uživatele, který příspěvek zadal nebo jeho přezdívku, pokud příspěvek zadal neregistrovaný návštěvník portálu. Pokud zadal příspěvek registrovaný uživatel, vkládá se do tabulky i jeho přezdívka pro případ, že by byl z databáze odstraněn, ale jeho příspěvek zůstane uložen.

Poslední dvě pole nazvané *ref* a *refId* zařadí příspěvek do správné diskuze. Pole *refId* obsahuje číslo nadřazeného příspěvku, pokud není vyplněno, jedná se o příspěvek hlavního vlákna diskuze. Pole *ref* má přednastaveno několik řetězcových konstant, které zařadí příspěvek do správné diskuze a to jsou „*\_public\_*“, „*\_members\_*“ a „*\_private\_*“. Tyto konstanty zařazují příspěvky do veřejné, členské a soukromé diskuze. U soukromé diskuze se mění význam sloupce *refId* tak, že neudrzuje vazbu na nadřazený příspěvek, ale vazbu na uživatele jemuž je příspěvek určen.

Zařazení příspěvků k ostatním diskuzím se provede vhodně zvolenými řetězci a to konkrétně řetězec „*\_playlist\_*“ pro playlist a „*\_action,x\_*“ pro zařazení příspěvku k akci, kde „*x*“ je identifikační číslo akce.

Za zmínku stojí také sloupec *state*, který udržuje informaci o tom, zda je příspěvek skrytý či ne. Tato funkcionality umožní vyřadit celé vlákno příspěvků a poté je znovu obnovit.

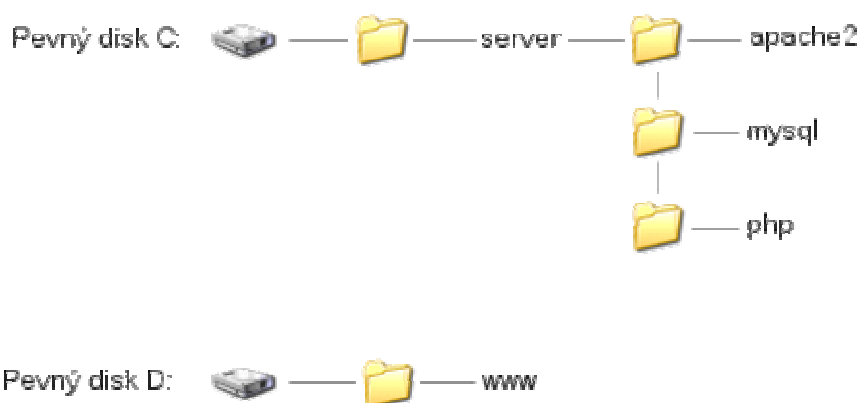
## 4 Instalace portálu

Kapitola se věnuje zprovoznění portálu na cílovém stroji. V úvahu se bere instalace na běžný hosting, který disponuje odpovídajícím programovým vybavením a disponuje alespoň základním administračním rozhraním umožňujícím operace s databázovým serverem MySQL, nebo je k dispozici složitější správce typu PhpMyAdmin. Postupně bude rozebrána i instalace a zprovoznění vlastního webového serveru založeného na freewarové trojici Apache — PHP — MySQL. Jako cílová platforma, pro kterou bude instalace webového serveru popsána, se předpokládá operační systém Windows.

### 4.1 Příprava

Nejprve je nutné připravit si prostor na stroji, na němž bude server provozován. Předpokládejme schéma, kdy na počítači jsou dva logické či fyzické disky. Jeden (c:\) je používán jako systémový a tedy je vhodné umístit na něj webový server. Druhý disk (d:\) je potom používán jako datový sklad, kde umístěna samotná aplikace.

Na systémovém disku vytvoříme adresář *server* a v něm podadresáře *apache2*, *php* a *mysql*. Na datovém disku vytvoříme adresář *www*. Struktura adresářů je zobrazena na obrázku 4.1 a slouží pouze jako referenční model, který je možné uzpůsobit jiným potřebám.



Obrázek 4.1: Referenční adresářová struktura pro webservice.

## 4.2 Instalace serveru Apache

Instalaci webového serveru Apache [5] usnadní instalační program s průvodcem instalací Microsoft Windows Installer (msi). Spuštění instalace se provede spuštěním souboru instalátoru *apache\_2.0.63-win32-x86-no\_ssl.msi*. Po spuštění instalace se zobrazí uvítací dialog a posunutím na další dialogy, kde jsou licenční údaje, údaje o verzi instalace a verzi serveru, se zobrazí dialogové okno s nastavením serveru: název domény, název serveru, e-mailová adresa administrátora serveru a způsob spuštění a provozu webového serveru. Tyto údaje je nutné vyplnit. Pro provoz lokálního serveru lze uvést smyšlené údaje, ovšem pro provoz v síti je třeba nastavit přesné hodnoty a to následující:

```
domain name:          kapela-marshal.cz
server name:          www.kapela-marshal.cz
admin e-mail address: admin@kapela-marshal.cz
```

Poslední položkou dialogového okna je nastavení způsobu spouštění serveru. To nastavíme na první nabízenou volbu a sice spouštění serveru jako servis Windows s nasloucháním příchozího spojení na portu 80, pro všechny uživatele.

V dalším dialogovém okně zvolíme typickou instalaci, protože není třeba explicitního nastavení serveru. Dále následuje dialog s možností zvolit cílové umístění serveru. Protože zvolený adresář v tomto dialogovém okně považuje instalační program za programovou skupinu *Apache Group* a v zájmu dodržení výše stanovené adresářové struktury, je nutné nastavit cílové umístění na adresář *c:\server*. Instalační program pro instalaci serveru použije složku *Apache2*. Posunem na další dialogové okno se spustí samotná instalace a po ní se instalační program rozloučí závěrečným dialogovým oknem.

Zda instalace proběhla v pořádku lze zjistit dvěma způsoby. Jednak to umožní program *Apache Monitor*, který je přístupný na systémové liště Windows (tray) a funguje jako ovládací program serveru. Druhým a dokonalejším způsobem je využití služby serveru. V libovolném prohlížeči zadáme adresu *http://localhost* nebo přímo lokální IP adresu *http://127.0.0.1* a server odpoví zobrazením webové stránky s textem „Funguje to, na tomto serveru běží Apache“.

## 4.3 Instalace interpretu PHP

Na rozdíl od serveru Apache balík PHP [7] neobsahuje instalační program a je tedy nutné jej nainstalovat „ručně“. Vzhledem k tomu, že pro funkci interpretu stačí vnímat PHP jako balík knihoven, kterých bude využívat server Apache, postačí rozbalit obsah archívu *php-4.4.8-Win32.zip* do připraveného adresáře *c:\server\php*.

Pro správnou funkci PHP je nutné zkopírovat knihovnu *php4ts.dll*, která se po rozbalení balíku PHP nachází v adresáři *c:\server\php*, do adresáře kde se nachází spouštěcí soubor webservru (*apache.exe*), tedy do adresáře *c:\server\apache2\bin*.

Dále je nutné vytvořit výchozí nastavení PHP. To lze nejlépe provést zkopírováním souboru *php.ini.recommended* z adresáře *c:\server\php* do adresáře *c:\windows* (podle instalace Windows) a jeho přejmenováním na *php.ini*. Samotné nastavení PHP bude popsáno v kapitole Nastavení a oživení webservru.

## 4.4 Instalace databázového serveru MySQL

Instalátor MySQL [6] poskytuje kompletního průvodce instalací a nastavením. Po spuštění instalátoru *mysql-essential-4.1.12a-win32.msi* se zobrazí uvítací dialog. Následuje dialog s výběrem typu instalace. Opět z důvodu dodržení adresářové struktury zvolíme vlastní instalaci, tedy volba „Custom setup“. V následujícím dialogu ponecháme nastavení instalovaných komponent a pouze změníme cílovou složku instalace na *c:\server\mysql*. Další dialog nabídne souhrn nastavení instalačního programu a po té následuje vlastní instalace serveru. Po skončení instalace jsme vyzváni k registraci produktu. Tato registrace není ovšem nutná, přeskočíme ji tedy výběrem volby „skip sign up“. Instalační program se rozloučí posledním dialogem instalátoru, v němž nabídne možnost server bezprostředně po instalaci konfigurovat. Zatrhnutím volby „configure“ a ukončením instalátoru se otevře průvodce konfigurací serveru MySQL.

Na výběr jsou dvě úrovně konfigurace a to standardní a detailní. Detailní je určena především pro konfiguraci MySQL na stroji, kde již existuje nebo existovala instance databázového serveru MySQL. Předpokládejme první instalaci databázového serveru, tudíž postačí standardní konfigurace, — volba „standard configuration“. V dalším dialogovém okně zvolíme způsob, jakým databázový server na stroji poběží. Zvolíme možnost, kdy server poběží jako servis Windows (volba „install as a service“) a název serveru ponecháme na přednastaveném „MYSQL“. Automatické spuštění servisu zajistíme zatrhnutím volby „launch mysql automatically“. Posledním dialogovém okně nastavíme heslo správce databáze (root) a v rámci vyšší bezpečnosti nezatrhnutím volby „create anonymous account“ znemožníme vytvoření anonymního přístupu k databázovému serveru.

## 4.5 Nastavení a oživení webservru

Všechny součásti webového serveru jsou nainstalované a zbývá poslední nejsložitější část což je nastavení a oživení webservru. K hladkému běhu webservru chybí nastavení komunikace mezi Apache a PHP a také nastavení umístění obsahu a kódování WWW.

Nejprve provedeme nastavení serveru Apache modifikací souboru s nastavením *c:\server\apache2\conf\httpd.conf*, který obsahuje textové nastavení serveru ve formátu velmi



podobném formátu souborů „ini“. Před nastavením je na místě připomenout, že jako oddělovač adresářů se v tomto případě používá dopředné nikoliv zpětné lomítko.

Abychom dodrželi adresářovou strukturu serveru a tedy plánované oddělení systémových souborů od WWW nastavíme klíč „DocumentRoot“, který je nyní nastaven na hodnotu `c:\server\apache2\htdocs`:

```
DocumentRoot "d:/www"
```

Tím jsme nastavili adresář `d:\www` jako výchozí adresář WWW a je nutné změnit nastavení k jeho přístupu. To provedeme nalezením tagu `Directory`, který má atribut `c:\server\apache2\htdocs` a nastavením atributu na cestu k novému umístění kořenového adresáře WWW, tedy:

```
<Directory "d:/www"> ... </Directory>
```

Nyní zaregistrujeme PHP skripty a nastavíme vazbu serveru Apache na PHP interpret po přidáním klíčů „AddType“ a „LoadModule“:

```
AddType application/x-httpd-php .php .inc  
LoadModule php4_module "C:/server/php/sapi/php4apache2.dll"
```

Oba klíče se přidají do konfiguračního souboru vždy ke skupině stejných klíčů, které již v souboru jsou. Tím se zachová správné pořadí klíčů.

Nyní doplníme seznam souborů, které jsou serverem považovány za indexový soubor adresářů WWW. Do seznamu klíče „DirectoryIndex“ přibude hodnota „index.php“:

```
DirectoryIndex index.html index.html.var index.php
```

Zbývá nastavení výchozí znakové sady pomocí klíče „AddDefaultCharset“:

```
AddDefaultCharset WINDOWS-1250
```

Na řadě je nastavení PHP. Soubor `php.ini`, který jsme již dříve nakopírovali do adresáře instalace Windows, je ve formátu klasického „ini“ souboru. Důležité je podotknout, že jako oddělovač adresářů slouží v tomto případě zpětné lomítko. Nejdříve nastavíme cestu k adresáři s knihovnamí, které PHP může vyžít. Jedná se o klíč „extension\_dir“ v sekci „PHP“:

```
extension_dir = "c:\server\php\extensions"
```

Pro běh internetového portálu je dále potřeba povolit používání grafické knihovny a to buď odstraněním komentáře u příslušného klíče, nebo přidáním klíče:

```
extension = php_gd2.dll
```

Dále nastavíme výchozí znakovou sadu:

```
default_charset = "windows-1250"
```

Jako poslední zbývá nastavit službu SMTP. Předpokládejme, že na stroji již běží poštovní server a tuto službu umožňuje a bude naslouchat na adrese *localhost* na standardním portu 25. V sekci „mail function“ tedy nastavíme příslušné hodnoty:

```
SMTP = localhost
smtp_port = 25
```

Tím je nastavení PHP dokončeno.

Protože výchozí nastavení MySQL plnohodnotně postačuje, přistoupíme k oživení webservru. Kliknutím pravým tlačítkem na ikonu monitorujícího programu Apache v systémové liště Windows (tray) a výběrem volby „Open Apache monitor“ se otevře dialog, který umožňuje ovládat server Apache. Tlačítkem „Restart“ provedeme restart serveru a s tím se projeví všechny provedené změny v konfiguraci Apache i PHP.

V případě, že se server již nespustí a skončí dialogem s chybovým hlášením, pravděpodobně se jedná o překlep v konfiguračním souboru. V tomto případě, je v rámci odhalení chyby dobré, spustit proces Apache (*apache.exe*) z příkazové řádky, kde se po neúspěšném spuštění serveru vypíše konkrétní chybové hlášení a tím se odhalí klíč konfiguračního souboru, který způsobuje chybu.

Pro ověření správné funkce PHP stačí vytvořit soubor *index.php*, který bude obsahovat volání jediné funkce a to funkce *phpinfo*. Pokud PHP funguje správně, vygeneruje se stránka, která detailně popisuje nastavení PHP. Podobně lze ověřit správnou funkci databázového serveru MySQL zavoláním funkce *mysql\_get\_server\_info*, která vypíše verzi databázového serveru.

## 4.6 Instalace internetového portálu

Před instalací internetového portálu je potřeba nakopírovat všechny soubory internetového portálu do kořenového adresáře WWW, tedy do adresáře *d:\www*. Nyní vytvoříme databázi pro internetový portál. V adresáři instalace MySQL, se nachází konzola, umožňující připojení k databázovému serveru (*c:\server\mysql\bin\mysql.exe*). Po spuštění konzoly bude vyžadováno heslo administrátora databáze, které bylo zadáno při instalaci. Vytvoření databáze lze provést ručně nebo lépe spuštěním skriptu *install/db.sql* v adresáři internetového portálu. Pokud je potřeba vytvořit databázi na veřejném hostingu, vytváří se databáze zpravidla v administrační sekci, která náleží k hostingu. V této situaci je nutné nastavit znakovou sadu databáze na *cp1250* a řazení v databázi na *cp1250\_czech\_cs*.

K vytvoření tabulek je určen skript *install/install.php* v kořenovém adresáři WWW. Po jeho spuštění se zobrazí přihlašovací formulář, kde uvedeme adresu serveru (*localhost*), přihlašovací jméno a heslo k databázi a název databáze (*marshal*). Dále jsou ve formuláři pole pro vytvoření administrátora internetového portálu. Po odeslání formuláře se vytvoří tabulky a účet administrátora.

Nakonec je třeba nastavit připojení k databázi v souboru *index.php*. Nastavení pro provoz na lokálním serveru se provede pomocí objektu *MySQL* a metody *SetConf* takto:

```
$MySQL->SetConf(LOCALHOST, 'localhost', '', '', 'marshal');
```

Nastavení pro provoz na hostingu je provede podle vzoru:

```
$MySQL->SetConf(SERVER, 'server', 'uživatel', 'heslo', 'databáze');
```

Údaje potřebné k nastavení připojení se váží k hostingu a zpravidla jsou zadány při vytváření databáze.

## 5 Závěr

Vytvořením stávajícího portálu se podařilo splnit závazky stanovené specifikací portálu. Portál plnohodnotně splňuje potřebu hudební skupiny a to jak z pohledu veřejné WWW prezentace, tak i z pohledu jednotlivých členů skupiny coby editorů informací. Administrační rozhraní implementované přímo do portálu poskytuje přehlednou práci s údaji a nabízí možnosti interní komunikace členů kapely.

Stejně tak implementace enginu splnila požadavky na vlastnosti enginu kladené. Při implementaci byly vlastnosti enginu operativně rozšiřovány podle potřeb, které se vyvíjely až během vývoje. Lze říci, že vyvinutý engine může být použitelný i v jiných portálech s podobnou tematikou a to s minimální potřebou přepsání jeho částí.

Nynější zpracování portálu si do budoucna vynucuje rozšíření komfortu práce použitím JavaScriptových WYSIWYG editorů, které interaktivně umožní vkládání HTML značek. Zcela jistě by obstála možnost větší svobody v rámci řešení práv uživatelů vytvářením šablon skupin uživatelů.

Z pohledu enginu, na kterém je portál založen, je zřejmé, že je na místě rozšíření enginu tak, aby byly plně pokryty všechny možnosti jazyka HTML a tím se docílilo úplné izolace PHP kódu od HTML značek. Dále by bylo vhodné přenést vázané moduly na vyšší míru abstrakce, aby se docílilo vyšší univerzálnosti celku a tím možnosti jeho použití nezávisle na tématice, kterou výsledný portál zpracovává.

# Literatura

- [1] Bráza, J., *PHP 4 – Praktické příklady*, Praha, Grada publishing 2003.
- [2] Mach, J., *PHP pro úplné začátečníky*, Brno, Computer Press, 2003.
- [3] Kosek, J., *PHP – Tvorba interaktivních internetových aplikací*, Praha, Grada publishing, 1999.
- [4] Šimůnek, M., *SQL – Kompletní kapesní průvodce*, Praha, Grada publishing, 1999.
- [5] The Apache Software Foundation, <http://www.apache.org/> (prosinec2007).
- [6] MySQL, <http://www.mysql.com/> (prosinec2007).
- [7] PHP, <http://www.php.net/> (prosinec2007)
- [8] PHP Documentation, <http://cz.php.net/docs.php> (leden 2008)
- [9] MySQL Documentation, <http://dev.mysql.com/doc/>, (leden 2008)
- [10] Historie PHP, <http://php.tonnikala.org/manual/cs/history.php> (březen 2008)
- [11] Tvorba Webu, <http://www.tvorba-webu.cz/php/historie.php> (březen 2008)

# Seznam příloh

Příloha 1. Nastavení enginu

Příloha 2. Struktura databáze

Příloha 3. CD se zdrojovými kódy portálu a programovým vybavením webserveru

## Příloha 1: Nastavení enginu

Nastavení enginu je možno provést ve skriptu settings.php. Význam konstant je popsán v tabulce.

Konstanta	Význam
ACTUALIZED_FILE	Název souboru s datem poslední aktualizace.
ADMIN_EMAIL	E-mailová adresa administrátora.
AUTO_LOGOUT_TIME	Čas, po kterém dojde k automatickému odhlášení uživatele.
DISCARD_REGISTRATION	Čas po kterém jsou vyřazeni uživatelé z databáze, pokud nepotvrdili registraci.
COUNTER_TIME_OUT	Čas po, kterém se započte další přístup ze stejné IP adresy do počítačidla přístupů.
COOKIE_EXPIRES	Doba platnosti údajů uložených v COOKIES.
TOPICS_PER_PAGE	Počet příspěvků na stránku.
INQUIRIES_PER_PAGE	Počet anket na stránku.
INQUIRY_TIME_OUT	Doba, po kterou nelze znovu hlasovat v anketě ze stejné IP.
PLAYLIST_NEWS_TIME	Doba, po kterou se píseň v playlistu považuje na novinku.
THUMB_WIDTH	Šířka náhledu ve fotogalerii.
THUMB_HEIGHT	Výška náhledu ve fotogalerii.
THUMBS_PER_ROW	Počet náhledů v řadě fotogalerie.
THUMBS_PER_PAGE	Počet řad fotogalerie na stránce.
AUTONEWS_COUNT	Počet zobrazovaných automaticky zaznamenaných novinek.
NEWS_COUNT	Počet zobrazovaných novinek.

## Příloha 2: Struktura databáze

### ACTIONS

ID	INT	NOT NULL	PRIMARY KEY
DATE	DATE	NOT NULL	
TIME	TIME		
PLACE	VARCHAR(40)		
TYPE	VARCHAR(60)		

### INQUIRIES

ID	INT	NOT NULL	PRIMARY KEY
CREATED	DATETIME	NOT NULL	
DEADLINE	DATETIME		
CAPTION	VARCHAR(40)	NOTNULL	
LEGEND	TEXT		
COLOR	CHAR(1)		DEFAULT „B“

### OPTIONS

ID	INT	NOTNULL	PRIMARYKEY
INQUIRYID	INT	NOTNULL	
CAPTOIN	VARCHAR (60)	NOTNULL	

### VOTES

ID	INT	NOTNULL	PRIMARYKEY
INQUIRYID	INT	NOTNULL	
OPTIONID	INT	NOTNULL	
IP	VARCHAR(15)	NOTNULL	
VOTED	DATETIMA	NOTNULL	

### BLOCKS

ID	INT	NOTNULL	PRIMARYKEY
RANK	INT	DEFAULT 0	
SECTION	VARCHAR(20)	NOTNULL	
NAME	VARCHAR(80)		
CONTENT	TEXT		

### PAPERS

ID	INT	NOTNULL	PRIMARYKEY
DATE	DATE	NOTNULL	
NAME	VARCHAR(80)	NOTNULL	
CONTENT	TEXT		

### DOWNLOADS

ID	INT	NOTNULL	PRIMARYKEY
TYPE	CHAR (1)	DEFAULT ' '	
RANK	INT	DEFAULT 0	
CAPTION	VARCHAR (40)	NOTNULL	
LEGEND	VARCHAR (100)		
LINK	VARCHAR (100)	NOTNULL	

FORUM			
ID	INT	NOTNULL	PRIMARYKEY
CREATED	DATETIME	NOTNULL	
STATE	ENUM ('A', 'X')	DEFAULT 'A'	
USERID	INT	NOTNULL	
USERNICK	VARCHAR(32)	NOTNULL	
REF	VARCHAR(10)	NOTNULL	
REFID	INT		
CONTENT	TEXT	NOTNULL	
PICTVIEW			
ID	INT	NOTNULL	PRIMARYKEY
DIR	VARCHAR(40)	NOTNULL	
ACTIONID	TEXT		
VIEWS	INT	NOTNULL	DEFAULT 0
LEGEND	TEXT		
PICTURES			
ID	INT	NOTNULL	PRIMARYKEY
PICTVIEWID	INT		
FILENAME	VARCHAR(40)	NOTNULL	
FILESIZE	INT	NOTNULL	
RANK	INT	NOTNULL	
RATING	INT	DEFAULT 0	
VOTES	INT	DEFAULT 0	
VIEWS	INT	NOTNULL	DEFAULT 0
LEGEND	TEXT		
LOGS			
ID	INT	NOTNULL	PRIMARYKEY
INSERTED	DATETIME	NOTNULL	
NICK	VARCHAR(32)	NOTNULL	
IP	VARCHAR(15)	NOTNULL	
MESSAGE	VARCHAR(100)		
NEWS			
ID	INT	NOTNULL	PRIMARYKEY
STAMP	DATE	NOTNULL	
LEGEND	TEXT	NOTNULL	
AUTONEWS			
ID	INT	NOTNULL	PRIMARYKEY
STAMP	DATE	NOTNULL	
TABLENAME	VARCHAR(20)	NOTNULL	
ITEMID	INT	NOTNULL	
LINKS			
ID	INT	NOTNULL	PRIMARYKEY
TYPE	CHAR 1	DEFAULT ' '	
RANK	INT	DEFAULT 0	
CAPTION	VARCHAR(40)	NOTNULL	



LINK	VARCHAR (100)	NOTNULL	
IMAGELINK	VARCHAR (100)		
PLAYLIST			
ID	INT	NOTNULL	PRIMARYKEY
INSERTED	DATETIME	NOTNULL	
TYPE	ENUM('P', 'Z', 'L')	DEFAULT 'Z'	
BAND	VARCHAR (32)	NOTNULL	
SONG	VARCHAR (32)	NOTNULL	
COUNTER			
ID	INT	NOTNULL	PRIMARYKEY
IP	VARCHAR (15)		
STAMP	DATETIME		
TERMINALS			
ID	INT	NOTNULL	PRIMARYKEY
STATE	ENUM ('I', 'O', 'X')	DEFAULT 'I'	
STAMP	DATETIME	NOTNULL	
CAPTION	VARCHAR (40)	NOTNULL	
LEGEND	TEXT		
CHECKS			
ID	INT	NOTNULL	PRIMARYKEY
TERMINALID	INT	NOTNULL	
USED	INT	NOTNULL	
CHECKED	ENUM ('I', 'O', 'X')	DEFAULT 'I'	
USERS			
ID	INT	NOTNULL	PRIMARYKEY
NICK	VARCHAR (32)	NOTNULL	
PWD	VARCHAR (32)	NOTNULL	
LEVEL	ENUM ('U', 'M', 'A')	DEFAULT 'U'	
RIGHTS	VARCHAR (40)		
ONLINE	ENUM ('Y', 'N')	DEFAULT 'N'	
STATE	ENUM ('R', 'A', 'B', 'X')	DEFAULT 'R'	
IP	VARCHAR (15)		
NAME	VARCHAR (32)		
LASTNAME	VARCHAR (32)		
EMAIL	VARCHAR (100)	NOTNULL	
PUBLICEMAIL	ENUM ('Y', 'N')	DEFAULT 'N'	
CREATED	DATETIME	NOTNULL	
LASTLOGIN	DATETIME		
VISITS	INT UNSIGNED	DEFAULT 0	
ICQ	VARCHAR (9)		
WEB	VARCHAR (100)		